(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6240108号
(P6240108)
(45)発行日　平成29年11月29日(2017.11.29)　(24)登録日　平成29年11月10日(2017.11.10)

(51)Int.Cl.　F I
G06F 17/30 (2006.01)　G06F 17/30 419B

請求項の数 5　(全 16 頁)

(21)出願番号　特願2015-30999(P2015-30999)
(22)出願日　平成27年2月19日(2015.2.19)
(65)公開番号　特開2016-152031(P2016-152031A)
(43)公開日　平成28年8月22日(2016.8.22)
審査請求日　平成28年12月8日(2016.12.8)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町一丁目５番１号
(74)代理人　110002147
特許業務法人酒井国際特許事務所
(72)発明者　新井　淳也
東京都千代田区大手町一丁目５番１号　日本電信電話株式会社内
(72)発明者　塩川　浩昭
東京都千代田区大手町一丁目５番１号　日本電信電話株式会社内

審査官　吉田　誠

最終頁に続く

(54)【発明の名称】リオーダリング装置、リオーダリング方法およびリオーダリングプログラム

(57)【特許請求の範囲】
【請求項１】
頂点と頂点との結合状態を隣接行列で表すグラフの頂点の集合から、頂点と頂点との前記結合状態に基づいて階層構造をなすクラスタを抽出する抽出部と、
抽出された前記各クラスタ内の頂点に連続した頂点ＩＤを付与し、該頂点ＩＤを用いて前記グラフの隣接行列を構成する構成部と、
を備えることを特徴とするリオーダリング装置。
【請求項２】
前記抽出部は、クラスタ内の頂点間の結合性を示す指標を最適化する塩川法を用いて前記クラスタを抽出することを特徴とする請求項１に記載のリオーダリング装置。
【請求項３】
前記抽出部は、前記クラスタを全頂点の集合を根とした多分木構造とみなし、親としての前記クラスタから該クラスタに含まれる子としての前記クラスタの順に、頂点を１つ含む前記クラスタまで抽出することを特徴とする請求項１または２に記載のリオーダリング装置。
【請求項４】
リオーダリング装置が実行するリオーダリング方法であって、
抽出部が、頂点と頂点との結合状態を隣接行列で表すグラフの頂点の集合から、頂点と頂点との前記結合状態に基づいて階層構造をなすクラスタを抽出する抽出工程と、
構成部が、抽出された前記各クラスタ内の頂点に連続した頂点ＩＤを付与し、該頂点Ｉ

Ｄを用いて前記グラフの隣接行列を構成する構成工程と、
を含んだことを特徴とするリオーダリング方法。
【請求項５】
頂点と頂点との結合状態を隣接行列で表すグラフの頂点の集合から、頂点と頂点との前記結合状態に基づいて階層構造をなすクラスタを抽出する抽出ステップと、
抽出された前記各クラスタ内の頂点に連続した頂点ＩＤを付与し、該頂点ＩＤを用いて前記グラフの隣接行列を構成する構成ステップと、
をコンピュータに実行させるためのリオーダリングプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、リオーダリング装置、リオーダリング方法およびリオーダリングプログラムに関する。
【背景技術】
【０００２】
情報検索や推薦、広告等の様々な場面で、ソーシャルグラフ、Ｗｅｂグラフ等のグラフを分析して有用な知識を発見している。一方、グラフの大規模化に伴って、分析に要する時間が増大するため、分析処理の高速化が求められている。
【０００３】
従来、ソーシャルグラフ、Ｗｅｂグラフなどの分析には、ＰａｇｅＲａｎｋ、Ｐｅｒｓｏｎａｌｉｚｅｄ　ＰａｇｅＲａｎｋ（ＰＰＲ）、Ｒａｎｄｏｍ　Ｗａｌｋ　ｗｉｔｈ　Ｒｅｓｔａｒｔ、ＨＩＴＳ（Hyperlink-Induced Topic Search）などの手法が知られている。これらの分析手法では、疎行列と密ベクトルとの乗算（以下、ＳｐＭＶ（Sparse Matrix-Vector multiplication）と称する。）が繰り返し実行される。
【０００４】
例えば、ＰＰＲは、グラフの重み付き隣接行列Ｗおよび各頂点への重みを表す列ベクトルｑから各頂点の重要度ｘを求める分析手法である。ｘは、次式（１）を収束するまで繰り返し計算することにより求められる。
【０００５】
【数１】

$$\boldsymbol{x}_{k+1} = cW\boldsymbol{x}_k + (1-c)\boldsymbol{q} \qquad \cdots (1)$$

ただしcは減衰比
【０００６】
ここで、一般に減衰比ｃは０．８５である。また、一般に隣接行列は疎行列となるため、Wx_kはＳｐＭＶである。上記の他の分析手法においても同様に、ＳｐＭＶが繰り返されるため、分析処理の高速化にはＳｐＭＶの高速化が重要である。
【０００７】
情報処理装置のＣＰＵ（Central Processing Unit）がＳｐＭＶを実行する際、局所性の低いメモリアクセスを行うことが知られている。例えば、次式（２）に示すｎ×ｎ疎行列Ａに対するＳｐＭＶである$Ax_{k+1}=Ax_k$の計算方法において、N_A（ｉ）は疎であることから、x_k［ｊ］に対するアクセスの局所性は低い。
【０００８】
【数２】

$$\boldsymbol{x}_{k+1}[i] = \sum_{j \in N_A(i)} A[i,j]\,\boldsymbol{x}_k[j] \ \text{ for all } \ i \in [0, n-1] \qquad \cdots (2)$$

ただし$N_A(i)$はAのi行目おいて非零要素が存在する列番号の集合
【０００９】

このような局所性の低いメモリアクセスパターンではキャッシュヒット率が低く、レイテンシの大きいメインメモリへのアクセスが頻発するため、処理速度が低下しやすい。また、今日一般的に使われているマルチコアＣＰＵでは、複数のコアが１つのメモリ帯域を共有するため、メモリアクセスが頻発すると、並列処理時に共有するメモリ帯域が飽和してスケーラビリティが低下する。そこで、ＳｐＭＶのメモリアクセスの局所性を向上させる技術が期待されている。

【００１０】

ＳｐＭＶのメモリアクセスの局所性を向上させるため、リオーダリングを行うことが効果的であることが知られている。リオーダリングとは、疎行列の行と列とを並べ替えてバンド幅を削減することを意味する。ここで、バンド幅を削減するとは、疎行列のｉ行目に存在する非零要素の分布をｉ列目付近に集中させることを意味する。この場合、上記式（２）においてｉ行目とその近傍のｉ’行目との間でｘに対する重複アクセスが増加して局所性が高まる。ここで重複アクセスは次式（３）で表される。

【００１１】

【数３】

$$N_A(i) \cap N_A(i') \qquad \cdots (3)$$

【００１２】

また、リオーダリングとは、隣接行列Ａをもつグラフにおいて、頂点ＩＤがｉである頂点に隣接する頂点に、ｉに近い頂点ＩＤを振り直すことに相当する。このリオーダリングによって隣接する頂点のデータをメモリ上で近傍に配置し直すと、幅優先探索（ＢＦＳ（Breadth First Search））等のＳｐＭＶを含まないグラフ処理でも局所性が向上することが確認されている。

【００１３】

ところで、ソーシャルグラフやＷｅｂグラフ等の実世界のグラフは、互いに密に接続し合った頂点の集合であるクラスタを多数内包し、高いクラスタ性を示すことが知られている。

【００１４】

ここで、グラフのエッジは隣接行列において非零要素で表されるので、クラスタ内の頂点に連続した頂点ＩＤを与えると、疎行列の対角線上に密な非零要素の領域を作り出すことができる。例えば、グラフから抽出された２つのクラスタC_1，C_2のそれぞれに含まれる頂点に連続した頂点ＩＤを与えた場合、クラスタ内のエッジすなわち非零要素は密である一方、クラスタ間のエッジである非零要素は疎である。したがって、非零要素の分布は、図１１に例示するように、非零要素が密である対角線に沿った濃色領域と、それ以外の非零要素が疎である白色領域とで表される。

【００１５】

このように疎行列の対角線上に密に非零要素を分布させれば、メモリアクセスの局所性を向上させることができる。そこで、グラフからクラスタを抽出する方法（ＣＮＭ法、非特許文献１参照）によって抽出されたそれぞれのクラスタについて、クラスタに含まれる頂点に連続した頂点ＩＤを与えるリオーダリング方法（ＣＯＭ、非特許文献２参照）が知られている。

【００１６】

また、実世界のグラフは階層的なクラスタ構造を持つことが知られている。例えば、図１２にソーシャルグラフを可視化して例示するように、丸で囲んで示すクラスタ内でもエッジの粗密に差がある。このように、階層的なクラスタ構造を持つ場合、クラスタの中からさらにクラスタを抽出することができる（塩川法、非特許文献３参照）。

【００１７】

そこで、クラスタ内のクラスタが内包する頂点に連続した頂点ＩＤを付与すれば、非零要素を疎行列の対角線上により密に分布させることができ、メモリアクセスの局所性を向上させることができる。例えば、図１３に例示するように、クラスタC_1に内包されるサ

ブクラスタC_{11}、C_{12}およびC_{13}と、クラスタC_2に内包されるサブクラスタC_{21}およびC_{22}とが、図１１のクラスタC_1およびC_2に比較して、疎行列の対角線上により密に分布していることがわかる。

【先行技術文献】

【非特許文献】

【００１８】

【非特許文献１】A.Prat-Perez, D.Dominguez-Sal, and J.L.Larriba-Pey、"Social Based Layouts for the Increase of Locality in Graph Operations."、In Proceedings of the 16th International Conference on Database Systems for Advanced Applications Volume Part I, DASFAA 2011、Springer-Verlag Berlin Heidelberg 2011、2011年、p.558-569

【非特許文献２】A.Clauset, M.E.J.Newman, and C.Moore、"Finding community structure in very large networks"、Physical Review E、2004年、70(6):66111

【非特許文献３】H.Shiokawa, Y.Fujiwara, and M.Onizuka、"Fast Algorithm for Modularity-Based Graph Clustering"、In Proceedings of the Twenty-Seventh AAAI Conference on Articial Intelligence (AAAI 2013)、2013年、p.1170-1176

【発明の概要】

【発明が解決しようとする課題】

【００１９】

しかしながら、従来のリオーダリング方法では、階層的なクラスタ構造が考慮されておらず、メモリアクセスの局所性の向上に限界があった。

【００２０】

また、従来のリオーダリング方法に用いられるＣＮＭ法は、十分な処理速度が得られず大規模なグラフへの適用が困難である。例えば、ＣＮＭ法によれば、約８０００万エッジ程度の規模のグラフからクラスタを抽出する処理が１日で完了できなかったとの報告例がある（非特許文献３参照）。

【００２１】

本発明は、上記に鑑みてなされたものであって、大規模なグラフの隣接行列のリオーダリングを行うことを目的とする。

【課題を解決するための手段】

【００２２】

上述した課題を解決し、目的を達成するために、本発明に係るリオーダリング装置は、頂点と頂点との結合状態を隣接行列で表すグラフの頂点の集合から、階層構造をなすクラスタを抽出する抽出部と、抽出された前記各クラスタを代表する頂点に連続した頂点ＩＤを付与し、該頂点ＩＤを用いて前記グラフの隣接行列を構成する構成部と、を備えることを特徴とする。

【発明の効果】

【００２３】

本発明によれば、大規模なグラフの隣接行列のリオーダリングを行うことができる。

【図面の簡単な説明】

【００２４】

【図１】図１は、本発明の一実施形態に係るリオーダリング装置の概略構成を示す模式図である。

【図２】図２は、本実施形態の階層的クラスタリング処理を説明するための図である。

【図３】図３は、本実施形態の頂点ＩＤ生成処理手順を例示する説明図である。

【図４】図４は、本実施形態の頂点ＩＤ生成処理手順を例示するフローチャートである。

【図５】図５は、本実施形態の塩川法が適用された階層的クラスタリング処理手順を例示する説明図である。

【図６】図６は、本実施形態の塩川法が適用された階層的クラスタリング処理手順を例示

するフローチャートである。
【図７】図７は、本実施形態の頂点の集約処理手順を例示する説明図である。
【図８】図８は、本実施形態の頂点の集約処理手順を例示するフローチャートである。
【図９】図９は、塩川法が適用された階層的クラスタリング処理における集約を説明するための説明図である。
【図１０】図１０は、リオーダリングプログラムを実行するコンピュータを示す図である。
【図１１】図１１は、クラスタ内の頂点に連続した頂点ＩＤを与えた場合の非零要素の分布を説明するための説明図である。
【図１２】図１２は、階層的なクラスタ構造を例示する模式図である。
【図１３】図１３は、クラスタ内の頂点に連続した頂点ＩＤを与えた場合の非零要素の分布を説明するための説明図である。
【発明を実施するための形態】
【００２５】
以下、図面を参照して、本発明の一実施形態を詳細に説明する。なお、この実施形態により本発明が限定されるものではない。また、図面の記載において、同一部分には同一の符号を付して示している。
【００２６】
［リオーダリング装置の構成］
まず、図１を参照して、本実施形態のリオーダリング装置の構成について説明する。リオーダリング装置１は、ワークステーションやパソコン等の汎用コンピュータで実現され、入力部２と、出力部３と、記憶部４と、制御部５とを備える。
【００２７】
入力部２は、電源スイッチおよび入力キーなどの入力デバイスを用いて実現され、操作者による入力操作に対応して、制御部５に対して各種指示情報を入力する。
【００２８】
出力部３は、液晶ディスプレイなどの表示装置、プリンターなどの印刷装置、情報通信装置などによって実現される。
【００２９】
記憶部４は、ＲＡＭ（Random Access Memory）、フラッシュメモリ（Flash Memory）等の半導体メモリ素子、または、ハードディスク、光ディスク等の記憶装置によって実現され、グラフデータ４１とリオーダリング結果４２とを格納する。グラフデータ４１は、リオーダリング装置１のリオーダリング処理の対象となるグラフの頂点の集合等の情報である。リオーダリング結果４２は、リオーダリング処理により出力される、例えばグラフの隣接行列等の情報である。なお、記憶部４は、ＬＡＮやインターネットなどの電気通信回線を介して制御部５と通信する構成でもよい。
【００３０】
制御部５は、処理プログラム等を記憶したメモリおよび処理プログラムを実行するＣＰＵ等を用いて実現され、抽出部５１および構成部５２として機能する。抽出部５１は、頂点と頂点との結合状態を隣接行列で表すグラフの頂点の集合から、階層構造をなすクラスタを抽出し、構成部５２が、抽出された各クラスタ内の頂点に連続した頂点ＩＤを付与し、該頂点ＩＤを用いてグラフの隣接行列を構成する。以下に、各構成部によるリオーダリング処理について詳細に説明する。
【００３１】
［リオーダリング処理］
本実施形態のリオーダリング処理では、グラフデータ４１から抽出されるグラフG_Aが入力される。なお、説明の簡略化のため、連結なグラフのみを対象とする。非連結なグラフについては、連結な部分ごとにリオーダリング処理対象とすればよい。また、リオーダリング処理では、グラフの頂点に新たな頂点ＩＤを付与する写像σが出力される。写像σは次式（４）で表すことができる。

【００３２】
【数４】

$$\sigma : \mathbb{V} \to \mathbb{N}. \qquad \cdots (4)$$

ただし$\mathbb{V}$は頂点の集合，$\mathbb{N}$は自然数の集合

【００３３】
本実施形態のリオーダリング処理は、初期化処理と、階層的クラスタリング処理と、頂点ＩＤ付与処理とで構成される。初期化処理は、メモリアクセス関係を表現するグラフを作成する処理である。階層的クラスタリング処理は、階層的クラスタ構造からクラスタを抽出する処理である。頂点ＩＤ付与処理は、抽出された全階層の全クラスタに連続した頂点ＩＤを与える処理である。

【００３４】
［初期化処理］
構成部５２は、頂点間で発生するデータの参照回数すなわちメモリアクセス回数を、エッジの重みで表現する重み付き無向グラフG_Mを生成する。ここで、頂点ＩＤがｖである頂点が頂点ＩＤがｕである頂点のデータを参照するとは、上記式（２）において、x_{k+1}［ｖ］の算出にx_k［ｕ］の値を必要とすること、すなわち、$u \in N_A(v)$を意味する。G_Mは、次式（５）で表される。

【００３５】
【数５】

$$G_M = (\mathbb{V}, \mathbb{E}, w) \qquad \cdots (5)$$

【００３６】
入力されたG_Aの隣接行列をｎ×ｎ行列Ａ＝（A_{ij}）とする。この場合、G_Mの隣接行列Ｍは次式（６）～（８）で定義される。

【００３７】
【数６】

$$M = \mathrm{NODIAG}\left(\mathrm{BOOL}\,(A) + \mathrm{BOOL}\,(A)^T\right) \qquad \cdots (6)$$

ただし，

$$\mathrm{BOOL}(A)_{ij} = \begin{cases} 0 & (A_{ij} = 0) \\ 1 & (A_{ij} \neq 0) \end{cases} \qquad \cdots (7)$$

$$\mathrm{NODIAG}(A)_{ij} = \begin{cases} 0 & (i = j) \\ A_{ij} & (i \neq j) \end{cases} \qquad \cdots (8)$$

【００３８】
ここで、G_Aのエッジの重みとは関係なく、エッジが存在すればデータ参照が発生するため、ＢＯＯＬ関数によって隣接行列Ｍを二値行列に変換する。また、クラスタリング処理によって抽出されるクラスタの品質は、有向グラフより無向グラフの方がよいことが知られている。そこで、ＢＯＯＬ（Ａ）＋ＢＯＯＬ（Ａ）^{T}によりG_Mを無向グラフ化する。さらに、自己ループA_{ii}の非零要素は、常に対角線上にあってリオーダリングの際に考慮する必要がないことから、ノイズとならないようにＮＯＤＩＡＧによって除去される。

【００３９】
［階層的クラスタリング処理］
抽出部５１は、階層的クラスタリング処理により、図２に例示するように、階層ｎの入れ子状のクラスタ構造を抽出する。ここで、階層はｎが大きくなるほど階層が低くなるも

のとする。最も低い階層のクラスタは頂点を１つだけ含む。

【００４０】

階層的クラスタリング処理により、以下の情報が得られる。まず、次式（９）で表される、トップレベルすなわちｎ＝１の階層に存在するクラスタの集合が得られる。

【００４１】

【数７】

$$\mathfrak{C}^{(1)} = \left\{ C_1^{(1)}, C_2^{(1)}, \dots \right\} \subset \mathfrak{P}(\mathbb{V}) \qquad \cdots (9)$$

ただし $\mathfrak{P}(\mathbb{V})$ は $\mathbb{V}$ の冪集合, $\bigsqcup \mathfrak{C}^{(1)} = \mathbb{V}$

【００４２】

また、次式（１０）で表される、階層ｎのｉ番目のクラスタを構成するサブクラスタの集合が得られる。

【００４３】

【数８】

$$\mathfrak{S}\left(C_i^{(n)}\right) = \left\{ C_{j1}^{(n+1)}, C_{j2}^{(n+1)}, \dots \right\} \subset \mathfrak{P}(\mathbb{V}) \qquad \cdots (10)$$

ただし $C_i^{(n)} = \bigsqcup \mathfrak{S}\left(C_i^{(n)}\right)$

【００４４】

また、次式（１１）、（１２）に示す定義により、階層的クラスタ構造は、図２に例示するように、階層０のクラスタＣ$_1^{(0)}$を根、階層ｎのｉ番目のクラスタを親、階層ｎのｉ番目のクラスタのサブクラスタを子とする多分木構造となる。

【００４５】

【数９】

$$C_1^{(0)} = \mathbb{V} \qquad \cdots (11)$$

$$\mathfrak{S}\left(C_1^{(0)}\right) = \mathfrak{C}^{(1)} \qquad \cdots (12)$$

【００４６】

なお、階層的クラスタリング処理には、具体的に、前述したＣＮＭ法、塩川法の他、Multi-level Label Propagation等の周知の方法のいずれを適用してもよい。塩川法を適用した具体例については後述する。

【００４７】

［頂点ＩＤ付与処理］

構成部５２は、階層的クラスタリング処理により得られた階層０のクラスタＣ$_1^{(0)}$と上記式（１０）で得られるサブクラスタとを用いて、頂点に新しいＩＤを与える写像σを求める。この写像σにより全階層の全クラスタの頂点が連続した頂点ＩＤとなるように頂点ＩＤが与えられる。この場合、次式（１３）が成り立つ。

【００４８】

【数１０】

$$\sigma\left(C_i^{(n)}\right) = \left[\underline{\sigma}_i^{(n)},\ \underline{\sigma}_i^{(n)} + \left|C_i^{(n)}\right| - 1\right] \qquad \cdots (13)$$

ただし $\underline{\sigma}_i^{(n)} = \min\left(\sigma\left(C_i^{(n)}\right)\right)$

【００４９】

上記式（１３）を満たす写像σは、例えば、図３に例示するアルゴリズム１で定義される。図４は、図３に示すアルゴリズム１を制御部５が実行する場合の処理手順を示すフローチャートである。アルゴリズム１は、ＡＳＳＩＧＮ関数の定義（図３の１～９行目、図

４のステップＳ１～Ｓ６）とその呼び出し（図３の１０行目）とで構成される。

【００５０】

ＡＳＳＩＧＮ関数は、階層ｎのｉ番目のクラスタ$C_i^{(n)}$とそのクラスタに含まれる頂点に与える最小の頂点ＩＤ番号を与えられる。内部では、階層ｎのｉ番目のクラスタ$C_i^{(n)}$を根とする多分木に対して深さ優先探索を行いつつ、クラスタに含まれる頂点への頂点ＩＤの割り当てを行う。最後に、クラスタ内の頂点へ与えた頂点ＩＤのうち、最大のものに１を足した値を戻り値として返却する。

【００５１】

具体的に、まず階層ｎのｉ番目のクラスタ$C_i^{(n)}$が、葉すなわち頂点を１つだけ含むクラスタであるか否かを判定し（図３の２行目、図４のステップＳ１）、もし葉であれば（ステップＳ１，Ｙｅｓ）、内包される唯一の頂点の頂点ＩＤとして、最小の頂点ＩＤを付与する（図３の３行目、図４のステップＳ２）。そして、内包される頂点に与えた頂点ＩＤのうち最大のものに１を足して返却する（図３の４行目、図４のステップＳ３）。

【００５２】

階層ｎのｉ番目のクラスタ$C_i^{(n)}$が、葉でない場合には（ステップＳ１，Ｎｏ）、最小の頂点ＩＤを一時変数ｓとして保存して（図３の６行目）、階層ｎのｉ番目のクラスタ$C_i^{(n)}$の子要素に対して再帰的にＡＳＳＩＧＮ関数を適用する（図３の７～８行目、図４のステップＳ４～Ｓ５）。最初の子要素に対するＡＳＳＩＧＮ関数の呼び出しの際に一時変数ｓを与え、その戻り値を一時変数ｓに保存して次の子要素に与える。これにより、クラスタ間で重複なく連続した頂点ＩＤが割り当てられる。

【００５３】

そして、一時変数ｓを返却して（図３の９行目、図４のステップＳ６）、ＡＳＳＩＧＮ関数を抜けるように定義される。このように定義されたＡＳＳＧＩＮ関数の呼び出し（図３の１０行目）では、全頂点に対して頂点ＩＤを付与するため、第１引数として多分木の根である階層０のクラスタ$C_1^{(0)}$が与えられる。また、全頂点の中で最小の頂点ＩＤは０であることから、第２引数に０が与えられる。以上の処理により、上記式（１３）を満たす頂点ＩＤが付与される。

【００５４】

最後に、構成部５２は、写像σにより付与された頂点ＩＤを用いてグラフG_Mのリオーダリングされた隣接行列を構成し、リオーダリング結果４２に格納する。なお、リオーダリング結果４２には、生成された写像σが格納されてもよい。

【００５５】

［階層的クラスタリング処理の具体例］

本実施形態では、階層的クラスタリング処理に塩川法が適用される。塩川法は、Ｍｏｄｕｌａｒｉｔｙの最適化を行うクラスタリング方法のうち、現在最も高速なアルゴリズムの１つである。ここで、Ｍｏｄｕｌａｒｉｔｙとは、クラスタリング結果を示す指標であって、クラスタリング結果が良いほど、すなわちクラスタ内の頂点間の結合性が高くクラスタが明確に分離可能であるほど高い値となる。頂点ｕ，ｖを１つのクラスタに分類して２頂点からなるクラスタを作った場合、すなわちこの２頂点を集約した場合、Ｍｏｄｕｌａｒｉｔｙの上昇量ΔＱ（ｕ，ｖ）は、次式（１４）で定義される。

【００５６】

【数１１】

$$\Delta Q(u,v) = 2\left(\frac{w_{uv}}{2m} - \frac{d_u d_v}{(2m)^2}\right) \qquad \cdots (14)$$

ただし，w_{uv} はエッジ $\{u,v\}$ の重み，

m は全エッジの重みの和，

d_u, d_v はそれぞれ u と v の重み付き次数

【００５７】
塩川法では、まず、頂点を逐次集約することにより、分析処理の高速化が図られている。すなわち、Ｍｏｄｕｌａｒｉｔｙを最も上昇させる隣接頂点へ、各頂点を貪欲に集約させることにより、グラフの規模を徐々に縮小させる。ここで集約された頂点が１つのクラスタを代表する頂点となる。
【００５８】
また、塩川法では、上記の逐次集約を次数の小さい頂点から順に行うことにより、分析処理の高速化が図られている。逐次集約によってグラフの規模が小さくなるため、当初は高次数だった頂点も集約が進むにつれ次数が低下する。したがって、次数の小さい頂点から順に集約すれば、エッジの参照回数が減少し、分析処理が高速化する。
【００５９】
［塩川法が適用されたリオーダリング処理］
図５は、階層的クラスタリング処理に塩川法が適用されたリオーダリング処理のアルゴリズムであるアルゴリズム２を示す図である。また、図６は、制御部５が実行するアルゴリズム２の処理手順を示すフローチャートである。このアルゴリズム２は、初期化処理（図５の１～２行目、図６のステップＳ１１～Ｓ１２）、階層的クラスタリング処理（図５の３～１１行目、図６のステップＳ１３～Ｓ２０）、および頂点ＩＤ生成処理（図５の１２～１７行目、図６のステップＳ２１～Ｓ２６）とで構成される。本実施形態では、階層的クラスタリング処理に、一部（図５の８～９行目、図６のステップＳ１８～Ｓ１９）を除いて塩川法が適用されている。
【００６０】
初期化処理では、変数の初期化と上記式（３）によるグラフの変換とが行われる。階層的クラスタリング処理では、塩川法を実行しつつ頂点集約履歴情報が収集される。まず、次数順に頂点ｕが選択され（図５の３～４行目、図６のステップＳ１３～Ｓ１４）、Ｍｏｄｕｌａｒｉｔｙの上昇量ΔＱ（ｕ，ｖ）が最大となる隣接頂点ｖの探索が行われる（図５の５行目、図６のステップＳ１５）。
【００６１】
Ｍｏｄｕｌａｒｉｔｙが上昇する場合（図５の６行目、図６のステップＳ１６，Ｙｅｓ）には、頂点ｕ，ｖは同一クラスタに分類される。その場合、頂点の集約（図５の７行目、図６のステップＳ１７）、クラスタのメンバ頂点数μの更新（図５の８行目、図６のステップＳ１８）、および頂点集約履歴リストの更新（図５の９行目、図６のステップＳ１９）が行われる。
【００６２】
一方、Ｍｏｄｕｌａｒｉｔｙが上昇しない場合（図５の１０行目、図６のステップＳ１６，Ｎｏ）には、頂点ｕが階層１のトップレベルのクラスタ集合に追加される（図５の１１行目、図６のステップＳ２０）。
【００６３】
ここで、頂点の集約は、図７に例示する塩川法の定義に従ったアルゴリズム３により実行される。図８は、抽出部５１が実行する図７のアルゴリズム３の処理手順を示すフローチャートである。まず、頂点ｕと接続されているエッジの端点をｖへつなぎ替え（図７の１行目、図８のステップＳ３１）、つなぎ替えにより重複するエッジの重みの和をとる（図７の３行目、図８のステップＳ３３）。ここで、$\mathrm{SUBST}_{uv}(e)$は、エッジの端点として出現するｕをｖに置き換えたエッジを意味する。
【００６４】
図９は、塩川法の定義に従った頂点の集約の過程を例示する図である。図９において、最も低い階層（ａ）は、各クラスタがただ１つの頂点を含む初期状態を示す。階層（ｂ）では、頂点の集約が進むにつれ、次数が小さい順に頂点v_6，v_0，v_2，v_9，v_5が残ったとする。ここで、最初に頂点v_6が集約の対象となり、Ｍｏｄｕｌａｒｉｔｙの上昇量が最も大きい隣接頂点を探索する。それがv_2であったとすると、v_6はv_2に集約される。階層（ｃ）において、頂点v_0も同様に、頂点v_9に集約される。

【００６５】
階層（ｄ）において、v_2の集約先を探索するも、Ｍｏｄｕｌａｒｉｔｙを上昇させる隣接頂点が見つからなかったとする。その場合、v_2は集約されずトップレベルのクラスタに分類される。その後、階層（ｅ），（ｆ）にて頂点v_9，v_5に対して集約がなされる。最終的に、次式（１５）で表される頂点集約履歴リスト、次式（１６）で表されるトップレベルのクラスタ集合、および図５に例示される各クラスタのメンバ頂点数μが得られる。
【００６６】
【数１２】

$$\mathbb{H} = \{\ldots,(v_6,v_2),(v_0,v_9),(v_9,v_5)\}, \quad \cdots(15)$$

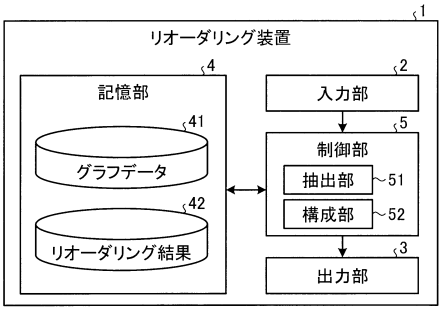
$$\mathbb{T} = \{v_2,v_5\} \quad \cdots(16)$$

【００６７】
図５および図６に戻る。頂点ＩＤ生成処理では、頂点集約履歴リストを利用して、頂点と新しい頂点ＩＤとの写像σが生成される。ここで、アルゴリズム２中のσ［ｖ］は、ｖによって代表されるクラスタに含まれる頂点に割り当てられる予定の頂点ＩＤのうち最小の頂点ＩＤを意味する。σは、図９の階層（ｇ）に示すトップレベルにあるクラスタについて最初に設定される（図５の１２～１４行目、図６のステップＳ２１～Ｓ２３）。
【００６８】
次に、頂点集約履歴リストを逆にたどり、図９の階層（ｆ）から階層（ａ）まで順に、集約された頂点を分解しつつσの値を更新する（図５の１５～１７行目、図６のステップＳ２４～Ｓ２６）。これにより、図９の階層（ａ）に示すように、各頂点が１つのクラスタになる初期状態まで分解され、σが各頂点と新しい頂点ＩＤとの対応を表す写像となる。
【００６９】
以上、説明したように、本実施形態のリオーダリング装置１では、抽出部５１が、頂点と頂点との結合状態を隣接行列で表すグラフの頂点の集合から、階層構造をなすクラスタを抽出し、構成部５２が、抽出された各クラスタ内の頂点に連続した頂点ＩＤを付与し、該頂点ＩＤを用いてグラフの隣接行列を構成する。
【００７０】
これにより、大規模なグラフの隣接行列について、行列バンド幅の削減率が高いリオーダリングを行うことができる。したがって、後続するグラフ処理やＳｐＭＶにおけるメモリアクセスの局所性が向上し、キャッシュヒット率が向上する。その結果、分析処理の処理速度が向上する。また、キャッシュヒット率向上の結果、メインメモリへのアクセス頻度が減少し、メモリ間帯域の飽和が緩和され、並列処理のスケーラビリティが向上する。また、塩川法を適用して高速に階層的クラスタリング処理を行うので、リオーダリング処理全体の処理時間が短縮される。したがって、本実施形態のリオーダリング装置１のリオーダリング処理によれば、大規模なグラフの隣接行列のリオーダリングを行って、大規模なグラフの分析処理におけるメモリアクセスの局所性を向上させることができる。
【００７１】
［他の実施形態］
［プログラム］
上記実施形態に係るリオーダリング装置１が実行する処理をコンピュータが実行可能な言語で記述したプログラムを作成することもできる。この場合、コンピュータがプログラムを実行することにより、上記実施形態と同様の効果を得ることができる。さらに、係るプログラムをコンピュータ読み取り可能な記録媒体に記録して、この記録媒体に記録されたプログラムをコンピュータに読み込ませて実行することにより上記実施形態と同様の処理を実現してもよい。以下に、リオーダリング装置１と同様の機能を実現するリオーダリングプログラムを実行するコンピュータの一例を説明する。

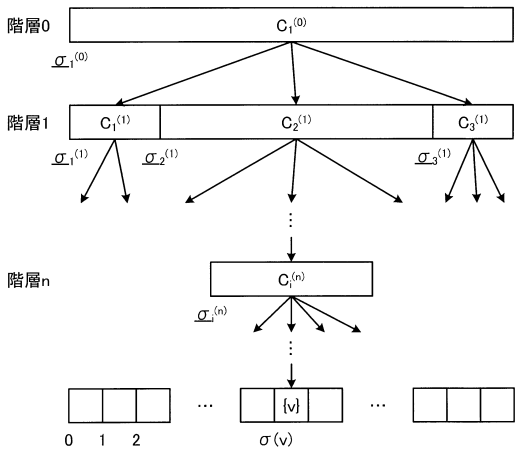
【００７２】
図１０に示すように、リオーダリングプログラムを実行するコンピュータ１０００は、例えば、メモリ１０１０と、ＣＰＵ１０２０と、ハードディスクドライブインタフェース１０３０と、ディスクドライブインタフェース１０４０と、シリアルポートインタフェース１０５０と、ビデオアダプタ１０６０と、ネットワークインタフェース１０７０とを有する。これらの各部は、バス１０８０によって接続される。
【００７３】
メモリ１０１０は、ＲＯＭ（Read Only Memory）１０１１およびＲＡＭ１０１２を含む。ＲＯＭ１０１１は、例えば、ＢＩＯＳ（Basic Input Output System）等のブートプログラムを記憶する。ハードディスクドライブインタフェース１０３０は、ハードディスクドライブ１０３１に接続される。ディスクドライブインタフェース１０４０は、ディスクドライブ１０４１に接続される。ディスクドライブ１０４１には、例えば、磁気ディスクや光ディスク等の着脱可能な記憶媒体が挿入される。シリアルポートインタフェース１０５０には、例えば、マウス１０５１およびキーボード１０５２が接続される。ビデオアダプタ１０６０には、例えば、ディスプレイ１０６１が接続される。
【００７４】
ここで、図１０に示すように、ハードディスクドライブ１０３１は、例えば、ＯＳ１０９１、アプリケーションプログラム１０９２、プログラムモジュール１０９３およびプログラムデータ１０９４を記憶する。上記実施形態で説明した各テーブルは、例えばハードディスクドライブ１０３１やメモリ１０１０に記憶される。
【００７５】
また、リオーダリングプログラムは、例えば、コンピュータ１０００によって実行される指令が記述されたプログラムモジュール１０９３として、ハードディスクドライブ１０３１に記憶される。具体的には、上記実施形態で説明したリオーダリング装置１が実行する各処理が記述されたプログラムモジュールが、ハードディスクドライブ１０３１に記憶される。
【００７６】
また、リオーダリングプログラムによる情報処理に用いられるデータは、プログラムデータ１０９４として、例えば、ハードディスクドライブ１０３１に記憶される。そして、ＣＰＵ１０２０が、ハードディスクドライブ１０３１に記憶されたプログラムモジュール１０９３やプログラムデータ１０９４を必要に応じてＲＡＭ１０１２に読み出して、上述した各手順を実行する。
【００７７】
なお、リオーダリングプログラムに係るプログラムモジュール１０９３やプログラムデータ１０９４は、ハードディスクドライブ１０３１に記憶される場合に限られず、例えば、着脱可能な記憶媒体に記憶されて、ディスクドライブ１０４１等を介してＣＰＵ１０２０によって読み出されてもよい。あるいは、リオーダリングプログラムに係るプログラムモジュール１０９３やプログラムデータ１０９４は、ＬＡＮ（Local Area Network）やＷＡＮ（Wide Area Network）等のネットワークを介して接続された他のコンピュータに記憶され、ネットワークインタフェース１０７０を介してＣＰＵ１０２０によって読み出されてもよい。
【００７８】
以上、本発明者によってなされた発明を適用した実施形態について説明したが、本実施形態による本発明の開示の一部をなす記述および図面により本発明は限定されることはない。すなわち、本実施形態に基づいて当業者等によりなされる他の実施形態、実施例および運用技術等は全て本発明の範疇に含まれる。
【符号の説明】
【００７９】
１　リオーダリング装置
２　入力部

3　出力部
4　記憶部
5　制御部
5 1　抽出部
5 2　構成部

【図１】



【図２】



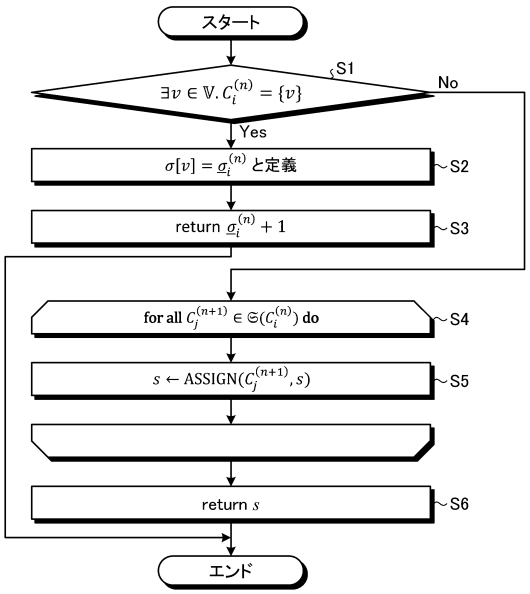
【図３】

アルゴリズム 1 ID の生成

入力: 階層 0 のクラスタ $C_1^{(0)}$，サブクラスタ集合 $\mathfrak{S}$
出力: 新しい ID への写像 $\sigma : \mathbb{V} \to \mathbb{N}$

function ASSIGN($C_i^{(n)}$, $\underline{\sigma}_i^{(n)}$)
　if $\exists v \in \mathbb{V}.\ C_i^{(n)} = \{v\}$ **then**
　　$\sigma[v] \leftarrow \underline{\sigma}_i^{(n)}$
　　return $\underline{\sigma}_i^{(n)} + 1$
　else
　　$s \leftarrow \underline{\sigma}_i^{(n)}$
　　for all $C_j^{(n+1)} \in \mathfrak{S}\left(C_i^{(n)}\right)$ **do**
　　　$s \leftarrow$ ASSIGN $\left(C_j^{(n+1)}, s\right)$
　　return s
ASSIGN $\left(C_1^{(0)}, 0\right)$

【図４】



【図５】

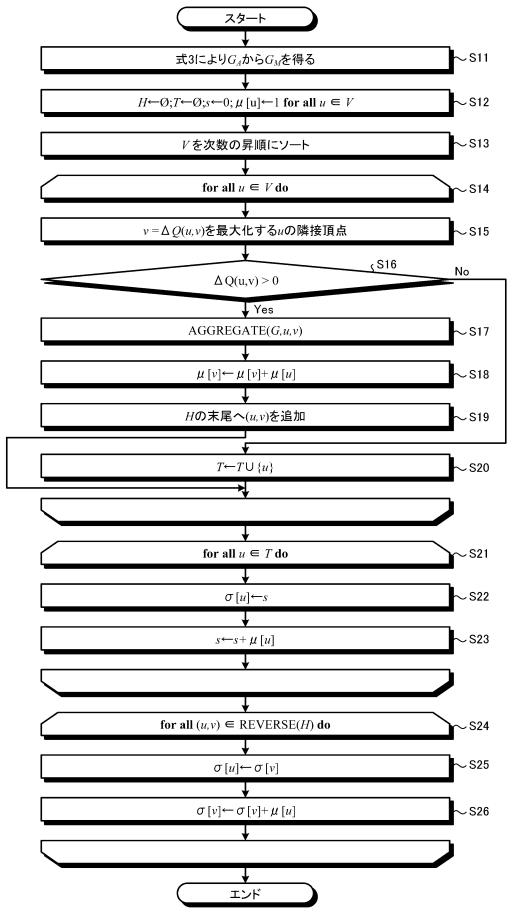
アルゴリズム 2 開発技術のアルゴリズム

入力: グラフ G_A

出力: 新しい頂点 ID を与える写像 $\sigma : \mathbb{V} \to \mathbb{N}$

式 3 により G_A から $G_M = (\mathbb{V}, \mathbb{E}, w)$ を得る
$\mathbb{H} \leftarrow \emptyset$; $\mathbb{T} \leftarrow \emptyset$; $s \leftarrow 0$; $\mu[u] \leftarrow 1$ **for all** $u \in \mathbb{V}$
$\mathbb{V}$ を次数の昇順にソート
for all $u \in \mathbb{V}$ **do**
    $v = \Delta\mathrm{Q}(u, v)$ を最大化する u の隣接頂点
    if $\Delta\mathrm{Q}(u, v) > 0$ **then**
        $\mathrm{AGGREGATE}(G, u, v)$
        $\mu[v] \leftarrow \mu[v] + \mu[u]$
        $\mathbb{H}$ の末尾へ (u, v) を追加
    else
        $\mathbb{T} \leftarrow \mathbb{T} \cup \{u\}$
for all $u \in \mathbb{T}$ **do**
    $\sigma[u] \leftarrow s$
    $s \leftarrow s + \mu[u]$
for all $(u, v) \in \mathrm{REVERSE}(\mathbb{H})$ **do**
    $\sigma[u] \leftarrow \sigma[v]$
    $\sigma[v] \leftarrow \sigma[v] + \mu[u]$

【図６】



【図７】

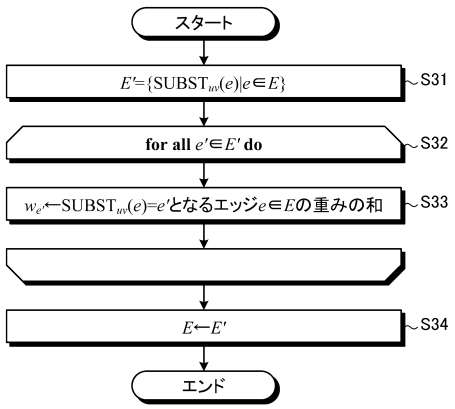
アルゴリズム 3 AGGREGATE 手続き

入力: グラフ $G_M = (\mathbb{V}, \mathbb{E}, w)$，頂点 u, v

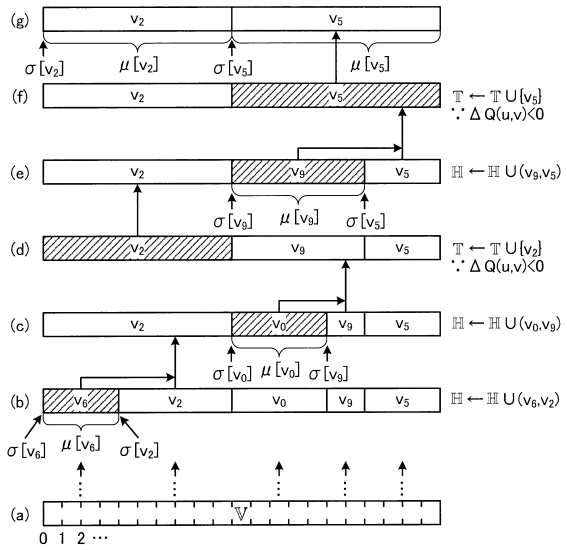
出力: u を v に集約したグラフ G_M

$\mathbb{E}' = \{\, \mathrm{SUBST}_{uv}(e) \mid e \in \mathbb{E} \,\}$ ▷ 集約後の新しいエッジ集合
for all $e' \in \mathbb{E}'$ **do**
    $w_{e'} \leftarrow \mathrm{SUBST}_{uv}(e) = e'$ となるエッジ $e \in \mathbb{E}$ の重みの和
$\mathbb{E} \leftarrow \mathbb{E}'$

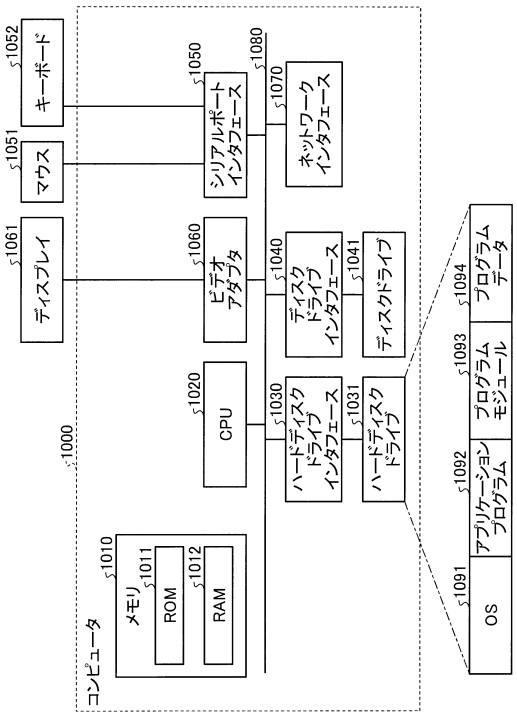
【図８】



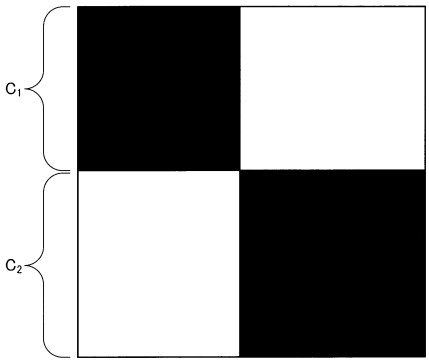
【図９】



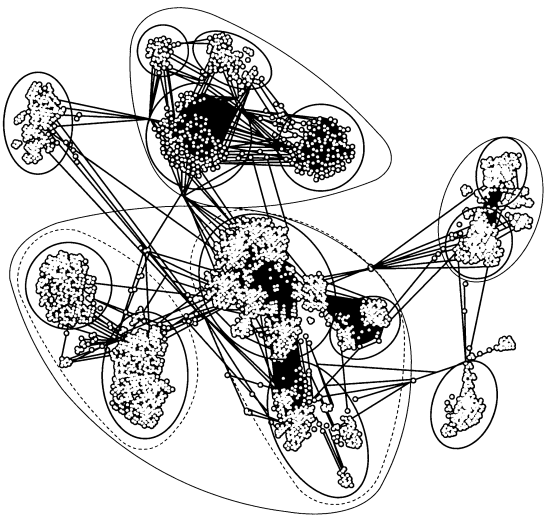
【図１０】



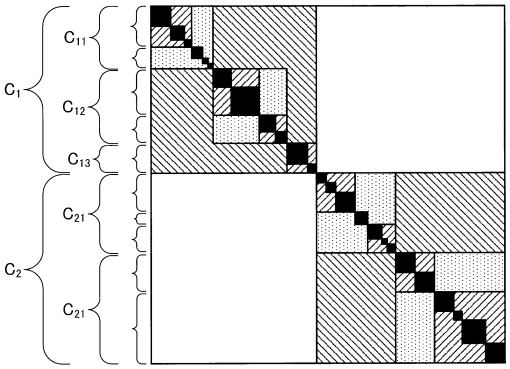
【図１１】



【図１２】



【図１３】



フロントページの続き

(56)参考文献 白山 晋, 大規模ネットワーク計算法に関して, 第１０回ＮＥシンポ合宿 予稿集 ［ＵＳＢ］, ２０１３年 9月 3日, １－１５２ページ
片瀬 弘晶, ＬｉｔｔｌｅＷｅｂ：類似ノード集約によるＷｅｂグラフ圧縮手法, 第2回データ工学と情報マネジメントに関するフォーラム－ＤＥＩＭ ２０１０－論文集, 日本, 電子情報通信学会データ工学研究専門委員会, ２０１０年 5月２５日, １－８ページ
飯田 恭弘, 企業におけるＡＩ研究の最前線, 人工知能, 日本, （一社）人工知能学会, ２０１４年 9月 1日, 第２９巻 第5号, ４７２－４７９ページ

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｆ １７／３０